



Universidad
Carlos III de Madrid
www.uc3m.es

Universidad Carlos III de Madrid
Escuela Politécnica Superior
Grado en Ingeniería Telemática

Trabajo Fin de Grado: Visualizaciones en la plataforma edX

Autor: David López Móriz
Tutor: Pedro José Muñoz Merino

Agradecimientos

Quisiera comenzar agradeciendo a mis padres todo el apoyo y la confianza que han depositado en mí siempre, no habría llegado hasta aquí sin ellos.

En segundo lugar, dar las gracias también a Laura por su apoyo y por aguantar conmigo todo el sufrimiento de estos años hasta llegar a este día.

Agradecer también a mi tutor, Pedro, por la oportunidad que me dio de realizar este Trabajo Fin de Grado y la paciencia que tuvo conmigo durante todo el proceso. Y también agradecer a José Antonio Ruipérez y a Jonathan Almodóvar, también personal de la UC3M, por su ayuda para poder arrancar con el proyecto.

Finalmente, no me quiero olvidar tampoco de mis amigos y mi familia, a los que agradezco su apoyo, y en especial, a Diego Lucero por sus consejos y su motivación, que hicieron que todo fuera más llevadero.

Índice general

Índice general.....	pág. 5
Índice de figuras.....	pág. 8
Índice de tablas.....	pág. 11
Glosario.....	pág. 13
CAPÍTULO 1 INTRODUCCIÓN.....	pág. 15
1.1 Introducción.....	pág. 16
1.2 Motivación y objetivos.....	pág. 18
1.3 Desarrollo del TFG.....	pág. 19
CHAPTER 2 INTRODUCTION.....	pág. 22
2.1 Introduction.....	pág. 23
2.2 Motivation and objective.....	pág. 25
2.3 TFG development.....	pág. 26
CAPÍTULO 3 TECNOLOGÍAS Y PLATAFORMAS UTILIZADAS.....	pág. 29
3.1 MOOC.....	pág. 30
3.2 edX.....	pág. 32
3.3 Instalación Open edX Platform.....	pág. 33
3.4 MySQL.....	pág. 35
3.5 MongoDB.....	pág. 36
3.6 D3.js.....	pág. 37
CAPÍTULO 4 ESTADO DEL ARTE.....	pág. 39
4.1 Learning Analytics y MOOCs.....	pág. 40
CAPÍTULO 5 DISEÑO E IMPLEMENTACIÓN DE LA SOLUCIÓN.....	pág. 44
5.1 Creación de datos.....	pág. 45
5.2 Consultas en MySQL.....	pág. 47
5.3 Consultas en MongoDB.....	pág. 52
CAPÍTULO 6 VISUALIZACIONES.....	pág. 62
6.1 Visualizaciones.....	pág. 63

CAPÍTULO 7 CONCLUSIONES Y TRABAJOS FUTUROS.....	pág. 76
7.1 Conclusiones.....	pág. 77
7.2 Trabajos futuros.....	pág. 79
CHAPTER 8 CONCLUSIONS AND FUTURE WORKS.....	pág. 81
8.1 Conclusions.....	pág. 82
8.2 Future works.....	pág. 83
CAPÍTULO 9 ANEXOS.....	pág. 85
9.1 Planificación.....	pág. 86
9.2 Medios técnicos empleados.....	pág. 89
9.3 Marco regulador.....	pág. 91
9.4 Análisis económico.....	pág. 93
BIBLIOGRAFÍA.....	pág. 95

Índice de figuras

Figura 1 – Logotipo de edX.

Figura 2 – edX logo.

Figura 3 – Conexión a la plataforma de edX.

Figura 4 – Conexión a MySQL y lista de bases de datos.

Figura 5 – Ejemplo de datos registrados en el tracking log.

Figura 6 – Código tiempo en los cursos.

Figura 7 – Código número de visitas.

Figura 8 – Código conexiones en el tiempo.

Figura 9 – Código número de alumnos por curso.

Figura 10 – Código última conexión.

Figura 11 – Código comentarios totales distribuidos en el tiempo.

Figura 12 – Código hilos iniciados distribuidos en el tiempo.

Figura 13 – Código respuestas distribuidas en el tiempo.

Figura 14 – Conexión a MongoDB y lista de bases de datos.

Figura 15 – Ejemplo de entrada en el documento contents.

Figura 16 – Código cantidad de hilos que inicia un usuario.

Figura 17 – Código cantidad de respuestas que hace un usuario.

Figura 18 – Código cantidad total de comentarios.

Figura 19 – Código perfil de usuario.

Figura 20 – Código media de comentarios que genera cada usuario.

Figura 21 – Código calidad de los comentarios.

Figura 22 – Código calidad de las preguntas.

Figura 23 – Código calidad de las respuestas.

Figura 24 – Código calidad media.

Figura 25 – Línea de código para elegir curso.

Figura 26 – Documentos que aparecen en localhost.

Figura 27 – Código php consultas MySQL.

Figura 28 – Código php consultas MongoDB.

Figura 29 – Línea código llamada librería D3.js.

Figura 30 – Ejemplo de código de llamada a los documentos php.

Figura 31 – Visualización calidad de los comentarios.

Figura 32 – Código html calidad de los comentarios.

Figura 33 – Visualización perfil de usuario.

Figura 34 – Código html perfil de usuario.

Figura 35 – Visualización número de conexiones por usuario.

Figura 36 – Código html número de conexiones por usuario.

Figura 37 – Visualización número de post totales.

Figura 38 – Código html número de post totales.

Figura 39 – Visualización conexiones en el tiempo.

Figura 40 – Visualización comentarios totales distribuidos en el tiempo.

Figura 41 – Visualización hilos iniciados distribuidos en el tiempo.

Figura 42 – Visualización respuestas distribuidas en el tiempo.

Figura 43 – Visualización calidad general de los comentarios.

Índice de tablas

Tabla 1 – Funcionamiento de los diferentes componentes.

Tabla 2 – Planificación TFG.

Tabla 3 – Diagrama de Gantt.

Tabla 4 – Presupuesto del TFG.

Glosario

- **AGPL** - Affero General Public License
- **BBDD** - Bases de Datos
- **CMS** - Course Management Systems
- **CPU** - Central Processing Unit
- **CSV** - Comma-Separated Values
- **DOM** - Document Object Model
- **HTML** - HyperText Markup Language
- **HTTP** - HyperText Transfer Protocol
- **JSON** - JavaScript Object Notation
- **LMS** - Learning Management Systems
- **LTS** - Long Term Support
- **MOOC** - Massive Open Online Course
- **PHP** - Hypertext Preprocessor
- **RAM** - Random Access Memory
- **SQL** - Structured Query Language
- **SVG** - Scalable Vector Graphics
- **TFG** - Trabajo Fin de Grado
- **UC3M** - Universidad Carlos III de Madrid

CAPÍTULO 1

INTRODUCCIÓN

1. Introducción

1.1 Introducción

En el mundo actual, globalizado, donde todo va dirigido cada vez más a internet y en el que los conocimientos son muy necesarios, surgió la educación en línea y a distancia para ofrecer un aprendizaje a cualquier persona en cualquier parte del mundo.

A partir de 1993, con el desarrollo de internet y los nuevos navegadores gráficos, este tipo de educación empezó a crecer de forma exponencial hasta nuestros días. Tanto es así, que se pueden encontrar opciones factibles para poder realizar estudios de bachillerato, universitarios y de especialización o postgrado, dando opciones de estudio a personas que por cualquier motivo no pueden asistir a cursos presenciales.

Algunas características de este tipo de educación son:

- El uso de una plataforma digital donde los alumnos pueden interactuar mientras transcurre el aprendizaje.
- El uso de material didáctico en diferentes formatos digitales (pdf, enlaces, videos, etc.).
- La inclusión de ejercicios para su realización, tanto en solitario como en grupo, por parte de los alumnos.
- La capacidad del alumno en relación a su autoaprendizaje.

A parte de pros, algunos mencionados anteriormente, también hay contras como, por ejemplo, la necesidad de tener unos conocimientos mínimos de tipo tecnológico y de internet, el aislamiento de los alumnos o la asincronía en la comunicación alumno-tutor o alumno-alumno.

Dentro de esta educación existen diferentes tipos de plataformas enfocadas de forma diferente:

- IBT (Instructional-Based Technology)
- CMC (Comunicación Mediada por Computador)
- LMS (Learning Management Systems)

- CMS (Course Management Systems)

Un ejemplo claro sobre el sistema LMS es la herramienta Moodle, muy difundida en el ámbito universitario y de software libre, que utiliza la Universidad Carlos III para su plataforma Aula Global.

Otra plataforma, siguiendo la línea de aprendizaje a distancia, pero con sus peculiaridades, llamada MOOC (Massive Open Online Course) o COMA en español (Curso Online Masivo Abierto), es la que nos interesa para la realización de este proyecto, y más concretamente, una organización que utiliza esta plataforma, edX, de la que hablaremos más adelante.



Figura 1 - Logotipo de edX

1.2 Motivación y objetivo

Viendo la evolución de todas estas herramientas y el aumento progresivo del número de alumnos en todas las plataformas, se entiende que tienen un futuro prometedor, pero muchas de ellas necesitan mejoras para seguir creciendo y mejorando la experiencia y la enseñanza de los alumnos y los tutores.

Para realizar esas mejoras, algo muy útil sería conocer diferentes datos sobre los usuarios y la interacción que hacen con la plataforma, y más concretamente con los cursos a los que estén apuntados, para poder cambiar lo que no se utiliza o lo que no esté bien, e incluso, para los tutores, tener datos de los alumnos y poder modificar los cursos en función de lo que reflejen los datos para mejorar la experiencia.

En este proyecto se utilizará la plataforma edX y se creará una funcionalidad que hasta ahora no tenía. Se realizará, en los cursos en línea, diferentes visualizaciones con respecto a los alumnos y el uso de los foros que hacen, desde el tiempo que los utilizan hasta las veces en las que entran, pasando por los comentarios que hacen y de qué tipo son. Todo ello se podrá visualizar en diferentes gráficas.

Toda esta información podrá ser utilizada en un futuro para crear una pestaña nueva en todos los cursos donde se pueda ver toda esa información útil, o aprovechar el conocimiento de las bases de datos para crear lo que llamaríamos gamificación, es decir, crear una serie de hitos a desbloquear por los usuarios como, por ejemplo, llegar a un número concreto de comentarios.

1.3 Desarrollo del TFG

La primera parte del Trabajo Fin de Grado es conocer la plataforma en la que se trabajará, en este caso edX. Ver cómo funcionan los cursos en línea, analizar las partes que tiene y observar con detenimiento el funcionamiento de los foros, que es la parte que más interesa para este proyecto.

A continuación, habría que instalar la aplicación de la plataforma diseñada para trabajar en local, creada para poder desarrollar la plataforma y poder probar mejoras y soluciones a problemas de manera más fácil.

Una vez instalado, hay que crear usuarios y cursos para poder tener datos a analizar. Esto es debido a que, al ser una aplicación que trabaja en local, no contiene ningún dato sobre cursos o alumnos existentes, por lo que hay que crear todo desde cero. La creación de los cursos esta apoyada en el estudio previo de la plataforma, donde se han estudiado las partes que contienen los cursos, además de cursos existentes en la propia web en los que te explican todos los pasos necesarios y posibles que se pueden hacer.

Lo siguiente a realizar, una vez creados los cursos y los usuarios, es realizar los cursos con cada usuario y generar actividad en los foros con todos ellos, utilizando todas las posibles opciones que da la plataforma.

El próximo paso del TFG es estudiar más en detalle el funcionamiento de edX, es decir, ver cómo se distribuye toda la información de los cursos, los alumnos y los foros. Todos estos datos están distribuidos en tres bases de datos diferentes y en diferentes lenguajes: para el contenido pedagógico se utiliza MongoDB, para los datos de usuario en entornos de desarrollo se utiliza SQLite y para los datos de usuarios en entornos de producción se utiliza MySQL.

Una vez analizadas las BBDD hay que crear las consultas necesarias para sacar los datos que queremos y ver si todas las consultas que queremos hacer son posibles, o faltan datos, o las tablas no son tan completas como se creía. Para todo esto último hay que tener los conocimientos necesarios en los lenguajes de MongoDB y MySQL.

Una vez tenemos las consultas tendremos que crear las visualizaciones de las mismas. Para ello, utilizaremos la librería de Javascript de D3.js, la cual nos permitirá representar los datos de manera dinámica vía web. Su funcionamiento se engloba con otras herramientas como php, html o apache, todas ellas necesarias para poder realizar los gráficos, por lo que hay que adquirir conocimientos en estos lenguajes también.

Una vez consigamos realizar las visualizaciones entraremos en la recta final del proyecto. Analizaremos los resultados obtenidos y procederemos a la realización de la memoria.

CHAPTER 2

INTRODUCTION

2. Introduction

2.1 Introduction

In a globalize world, where everything is achievable thanks to an easy and affordable use of internet, and where the knowledge is very useful, it started an era of distance and online education to offer to every student all around the world.

In 1993, with the internet and the new graphic browsers development, this kind of online education started an exponential growth until now. Nowadays it is achievable to everybody who cannot assist to a face-to-face classes, and therefore, is possible to course form High School Diplomas to University and Master's Degrees, the same as any other kind of supplementary courses or languages.

These are some of this online education's features:

- The use of a digital platform where all the students can interact while the course is open.
- The use of different instructional materials may be in a digital format, such as pdf formats, videos, and links.
- The inclusion of both individual and groups exercises.
- The student's capacity in relation with his self-learning.

As well as "pros", some of them mentioned before, there are also "cons", like for instance, the necessity of having some minimum technical or internet knowledge, the student's isolation or the asynchronous' communication between the student and the teacher, or student to student.

Inside this "online education" it exists different types of platforms:

- IBT (Instructional-Based Technology)
- CMC (Comunicación Mediada Por Computador)
- LMS (Learning Managment Systems)
- CSM (Course Managment Systems)

A clear example of LMS is the Moodle tool (widely distributed on the academic environment), which uses a free software and it is used by Carlos III University for its "Aula Global" platform.

Another platform, following the distance-learning line but with its own peculiarities, called MOOC (Massive Open Online Course) or COMA in Spanish (Curso Online Masivo Abierto), is the one that we are interested in for the realization of this project, and more in particular the organization which uses this platform, edX, that I will speak about it afterwards.



Figure 2 - edX logo

2.2 Motivation and objective

If we take a look at the evolution of these tools and the progressive increase number of students in all platforms, it is understood that they have a promising future, but many of them they need improvements to keep growing like the students and the teachers' experience and education.

To make that progress, it would be so useful to know some information from the users and their interaction with the platform, and more in particular with the courses they signed up, in order to change either what is not used or what it is wrong, or even for the teachers, to obtain some information so they can modify the courses for improvement proposes.

In this project it will be used edX platform and it will be created a function not used not so far. On the online courses, It will be displayed the student's use of the platform, from how long they stay on the platform until how many times they log in, the comments they write on the forums and what type of comments they do. All this its reflected on different charts.

2.3 TFG development

The first part of the End Of Grade is to know the platform we are going to work with, in this case edX. See how online courses work, analyze the parts it has and carefully observe the forums' functioning, which is the most important part for this project.

The next step, is to set up the platform's application designed to work in local, created to be able to develop the platform and to be able to test improvements and solutions to problems in an easier way.

Once installed, we have to create users and courses to be able to have data to analyze. This is because, being an application that works in local, does not contain any data about existing courses or students, so you have to create everything from scratch. The creation of the courses is supported by the previous study of the platform, where they have studied the parts that the courses' contains, in addition to existing courses in the web itself in which they explain all the necessary and possible steps that can be done.

The next thing to do, once the courses and users are created, is to conduct the courses with each user and generate activity on the forums with all of them, using all the possible options given by the platform.

The next step of the TFG is to study more in detail the operation of edX, that is, to see how to distribute all the courses, students and forums' information. All these data are distributed in three different databases and in different languages: MongoDB is used for the pedagogical content, SQLite is used for data in development environments and MySQL is used for user data in production environments.

Once the BBDD have been analyzed, it is necessary to create queries to obtain the data that we want, and see if all the queries we want to do are possible, or some data is missing, or the tables are not as complete as we thought. For all this, it is a must to have the necessary MongoDB and MySQL languages' knowledge.

Once we have the queries we will have to create the visualizations. To do this, we will use the D3.js' Javascript's library, which will allow us to represent the data dynamically via the web. Its operation is integrated with other tools like php, html or

apache, all of them are required to be able to create the graphics, so it is important to get the knowledge on these languages also.

Once we get the visualizations done we will enter the final line of the project. We will analyze the obtained results and proceed to the memorandum completion.

CAPÍTULO 3

TECNOLOGÍAS Y

PLATAFORMAS UTILIZADAS

3. Tecnologías y plataformas utilizadas

En este capítulo se explicarán la plataforma y las tecnologías sobre las que trabajaremos en este proyecto.

3.1 MOOC

Los MOOC son el acrónimo de Massive Online Open Courses, y no es otra cosa que la evolución de la educación abierta en internet. Gracias a esto cualquiera puede realizar un curso gratis en Harvard, Stanford, el MIT o cualquier universidad del mundo que se te pueda ocurrir sin necesidad de ir hasta sus instalaciones.

La historia de los MOOC empezó en el año 2008 con la creación del primer curso, “Connectivism and Connective Knowledge (CCK08)”, diseñado por George Siemens y Stephen Downes. Ante la acogida que tuvo el curso, se empezaron a crear más cursos abiertos llegando a cada vez más personas. El punto de inflexión de estos cursos llegó en 2011 gracias a un curso del profesor de Stanford, Sebastian Thrun, y su curso sobre inteligencia artificial, el cual atrajo a 160.000 estudiantes, y publicaciones en la prensa como en el New York Times. Todo ello contribuyó a la expansión de estos cursos en todo el mundo.

Hoy en día hay 33 plataformas de cursos MOOC repartidas por todo el mundo, donde universidades públicas y privadas, institutos, organizaciones sin ánimo de lucro, corporaciones y organizaciones internacionales comparten sus cursos de forma gratuita. Entre esas plataformas destacan Coursera, Udacity, LearnDash y edX, entre otras.

Las principales características de estos cursos son:

- Acceso libre y abierto: no se requiere pertenecer a ninguna institución y no hay que demostrar unos conocimientos previos.
- Acceso gratuito: el acceso a la plataforma y al contenido de los cursos es gratis, aunque las tutorías, evaluaciones, recursos bibliográficos o acreditaciones pueden estar sujetas a algún tipo de pago.
- Todo el curso se tiene que desarrollar en línea.
- Interacción en línea alumno-profesor y alumno-alumno.

- Sin limitación en las matriculaciones.
- Contenidos con licencia para permitir su reutilización.
- Estructura concebida para promover el aprendizaje autónomo.
- Evaluación de una serie de pruebas que acrediten el haber adquirido los conocimientos necesarios.

Todas estas características lo diferencian de los cursos en línea tradicionales, ya que estos son cerrados, no son gratuitos y hay un número limitado de alumnos, entre otras cosas.

3.2 edX

edX es una plataforma de cursos MOOC creada en mayo de 2012 por el Instituto Tecnológico de Massachusetts y la Universidad de Harvard, con la idea de que fuera universal y sin costes para propiciar la investigación y el aprendizaje.

El prototipo del primer curso se lanzó en diciembre de 2011 con el nombre de “Circuits and electronics from MIT” creado por el actual CEO de la plataforma, Anant Agarwal, que consiguió que se apuntaran a su curso 155.000 estudiantes de 162 países diferentes.

En la actualidad, 111 instituciones de todo tipo y de todos los rincones del mundo forman parte de la plataforma. 50 de ellos son socios fundadores, entre los que se encuentran instituciones como, por ejemplo, la Universidad de Pekín, la Universidad Nacional de Seúl, la Universidad Técnica de Munich, la Universidad Nacional Australiana o el Instituto de Tecnología de California. Otras 57 instituciones son miembros de la plataforma, entre los que se encuentran la Universidad Carlos III de Madrid, el Fondo Monetario Internacional, el Banco Interamericano de Desarrollo, Amnistía Internacional o Microsoft. Y por último, 4 “amigos” de la organización como la Fundación Bill y Melinda Gates. Todos ellos participan de una forma u otra en la plataforma.

Los números que hacen que edX sea una de las plataformas de cursos MOOC más conocidas son los más de 10 millones de estudiantes registrados y los cerca de 1.500 cursos que se ofrecen, entre los que se reparten más de 33 millones de suscripciones.

El funcionamiento de la plataforma es sencillo. El primer paso es registrarse en la plataforma, lo cuál es muy sencillo. Una vez registrado e iniciado sesión, se pueden buscar los cursos que queramos mediante filtros como palabras clave, nivel, idioma o temario. Una vez encontremos un curso que nos guste, nos apuntamos al curso y a partir de ahí, podremos empezar a ver el contenido, todo ello de manera gratuita. Si quieres conseguir un título específico de la institución que ofrece el curso, una vez acabado el mismo y habiéndolo superado, habría que pagar una cantidad que varía en función del curso y de la institución.

3.3 Instalación Open edX Platform

A continuación, se detallará el proceso de instalación de la plataforma local con la que se trabajará durante el TFG y se verán las diferentes instalaciones existentes.

Lo primero que se necesita es un sistema operativo Linux, ya sea utilizando un ordenador con Linux instalado o utilizando una máquina virtual en la que instalemos este sistema operativo. Lo siguiente es ver qué tipo de instalación queremos, ya que existen 4 diferentes con diferentes componentes de software entre ellas (Devstack, Fullstack, Analytics Devstack y Native). Para el caso que nos concierne utilizaremos un ordenador con Ubuntu 14.04 LTS e instalaremos Open edX Devstack, ya que, es la que mejor se adapta a nuestros objetivos.

Devstack es una instancia de Vagrant diseñada para el desarrollo local. Algunas de sus características son: permite descubrir y corregir los problemas de configuración del sistema al principio del desarrollo, simplifica ciertas configuraciones de producción para hacer el desarrollo más conveniente y utiliza el runserver de Django. En este caso, los componentes de software que se han instalado son los siguientes:

- The Learning Management System (LMS): sistema principal donde se accede a los cursos y a su contenido.
- Open edX Studio: sistema web donde se crean los cursos y se gestionan sus contenidos.
- Discussion Forums: sistema que permite el funcionamiento de los foros en los cursos creados.
- Open Response Assessments (ORA): sistema de evaluación de los cursos.
- E-Commerce: aplicación Django utilizada para administrar el catálogo de productos de edX y gestionar pedidos para esos productos.
- Programs
- Demonstration Open edX course: curso de introducción a edX
- Open edX Search: aplicación de Django que proporciona acceso a servicios de búsqueda desde aplicaciones de plataforma edX

El último paso, antes de entrar a la instalación en si, es instalar VirtualBox, Vagrant y un cliente NFS (Network File System), todo ello indispensable para el correcto funcionamiento de la plataforma.

Con todo esto ya podremos instalar la plataforma Devstack en el ordenador. La versión utilizada para el TFG es la Eucalyptus.1, lanzada el 26 de agosto del 2016, no siendo actualmente la más moderna. Todos los pasos de instalación vienen explicados perfectamente en los links que se apuntan en la bibliografía al final de la memoria, concretamente los puntos 6 y 7. Una vez realizado todo, y desde nuestro terminal, podremos acceder a la plataforma, desde donde trabajaremos lanzando los servidores para poder utilizar las herramientas LMS, Studio y Forum, y también para poder consultar las tablas de las diferentes bases de datos.

[illegible]

Figura 3 - Conexión a la plataforma de edX

3.4 MySQL

MySQL fue lanzado el 23 de mayo de 1995 en su primera versión. Actualmente es el segundo sistema relacional de gestión de bases de datos más utilizado del mundo, el cual usan sitios tan conocidos como Wikipedia, Google, Facebook, Twitter o Youtube.

Como su nombre indica, utiliza SQL (Structured Query Language) como lenguaje de acceso a la base de datos, ya que esta estandarizado para almacenar, actualizar y administrar una base de datos. Todo ello se programó en su mayoría en C# y C++. Su funcionamiento es sencillo, se utiliza una base de datos como servidor (MySQL) y los clientes se comunican para acceder a dicho servidor para consultar y/o modificar la información de dicha base de datos.

Las principales características de MySQL son:

- Compatibilidad: compatible con muchos sistemas operativos como Windows, Linux, MacOS, SunOS, NetBSD...
- Velocidad: programa de acceso de base de datos muy rápido.
- Claves foráneas: limitación referencial entre dos tablas.
- Replicación: proceso de copiar y mantener actualizados los datos en varios nodos de bases de datos, ya sean estos persistentes o no.
- Lenguajes de programación: gran número de APIs y librerías para el desarrollo de aplicaciones MySQL, programables en distintos lenguajes como C#, C++, Java, Python, PHP o Perl.

3.5 MongoDB

El desarrollo de esta base de datos empezó en 2007 y fue lanzado finalmente en 2009 bajo la licencia de código abierto AGPL (Affero General Public License), pero hasta marzo de 2011, con la versión 1.4, no se consideró una base de datos lista para su uso en producción.

MongoDB es un sistema de base de datos no relacional (NoSQL), orientado a documentos y desarrollado bajo el concepto de código abierto. Los datos no se guardan en tablas, sino en estructuras de datos en documentos con un esquema dinámico, de tipo BSON (Binary JSON (JavaScript Object Notation)). Esta disponible para los sistemas operativos Windows, Linux, OS X y Solaris, y es utilizado por empresas como MTV Network, Craigslist o Foursquare.

Las principales características de MongoDB son:

- Consultas Ad hoc: se pueden realizar búsquedas por campos, consultas de rangos y expresiones regulares.
- Indexación: se puede indexar cualquier campo en un documento.
- Replicación: soporta la replicación primario-secundario. El primario ejecuta y el secundario replica los datos.
- Balanceo de carga: se distribuyen los datos de forma horizontal entre múltiples máquinas utilizando “sharding”. Esto se utiliza para soportar despliegues de datos muy grandes y operaciones de alto rendimiento.
- Almacenamiento de archivos: puede ser utilizado como un sistema de archivos beneficiándose de otras características descritas aquí, como el balanceo de carga o la replicación.
- Agregación: proporciona un framework de agregación que permite realizar operaciones similares a las que se obtienen con el comando “GROUP BY” de SQL.
- Ejecución de JavaScript del lado del servidor.

3.6 D3.js

D3 (Data-Driven Documents) es una librería de Javascript que permite enlazar datos arbitrarios a un DOM (Modelo de Objetos de Documento), y a continuación, producir infogramas dinámicos e interactivos en navegadores web.

Lanzado inicialmente el 18 de febrero de 2011, D3 es la librería sucesora de Protovis. D3 resuelve el principal problema de estas librerías: la manipulación de documentos basados en datos. Embebido dentro de una página web de HTML, utiliza funciones Javascript predefinidas para seleccionar elementos, crea objetos SVG, y les da estilo o agrega información, efectos dinámicos o transiciones. Los datos se pueden añadir a objetos SVG fácilmente para su posterior visualización. Estos datos pueden estar contenidos en diversos formatos como JSON, CSV o GeoJSON, o como en el caso que nos concierne, podremos realizar consultas directamente a bases de datos.

Su uso ha sido variado, desde ser utilizado para desarrollar mapas mediante archivos GeoJSON y TopoJSON, hasta ser utilizado por medios de comunicación como The New York Times para generar infogramas.

Las principales características de D3 son:

- Flexibilidad y control total de las representaciones.
- Sobrecarga mínima.
- Extremadamente rápido: soporta grandes conjuntos de datos y comportamientos dinámicos.
- Reutilización de código.

CAPÍTULO 4

ESTADO DEL ARTE

4. Estado del arte

4.1 Learning analytics y MOOCs

Learning analytics es la medición, recopilación, análisis y presentación de datos sobre los estudiantes, sus contextos y las interacciones que allí se generan, con el fin de comprender el proceso de aprendizaje que se está desarrollando y optimizar los entornos en los que se produce. En español algunos autores la han denominado analítica de aprendizaje o análisis del aprendizaje.

Estos análisis sirven para ayudar a los usuarios (estudiantes, profesores, administradores) de un LMS a mejorar los resultados del aprendizaje o funciones dentro del sistema educativo.

Una definición de esta herramienta, aunque discutida, es que el “Learning Analytics es el uso de datos inteligentes, datos del alumno-producto, y modelos de análisis para descubrir las conexiones de la información y sociales, que se originan en un entorno digital para predecir y asesorar el aprendizaje de las personas.”

En contra de ella, se han hecho varias reflexiones:

- George Siemens: “Sirve como un concepto de introducción si utilizamos análisis como una estructura de soporte para los modelos de educación existente. En una implementación avanzada e integrada, se puede acabar con los modelos curriculares prefabricados.”
- Mike Sharkey: “En las descripciones del Learning Analytics hablamos de uso de datos para predecir el éxito.”
- Begoña Gros: “El análisis del aprendizaje consiste en la interpretación de un amplio espectro de datos producidos y recogidos de los estudiantes para orientar su producción académica, predecir acciones futuras e identificar elementos problemáticos. El objetivo es posibilitar que los profesores puedan adaptar de forma rápida y eficaz las estrategias educativas que más se adapten a sus alumnos.”

Existen varios métodos de análisis de aprendizaje: por contenido, por discurso, por disposición, por redes sociales, por estadística, por minería de datos y por visualización. Estas cuatro últimas son las que más atención han recibido en los últimos años, y en este TFG en particular, nos interesa la visualización de la información a través de imágenes para facilitar la comprensión y el análisis de datos.

Todo ello nos sirve para mejorar varios aspectos del aprendizaje (como decía Peter Drucker, “todo lo que se puede medir se puede mejorar”):

- Mejor toma de decisiones administrativas y asignación de recursos.
- Predicción de comportamientos de los estudiantes.
- Intervención de los profesores gracias al punto anterior.
- Personalización y adaptación de los contenidos.
- Empoderamiento de los estudiantes para seguir su progreso.
- Visualización de la información.

Esta información se podría extender muchísimo con todas las referencias que existen sobre este tema, como por ejemplo, el libro de Daniel Amo y Raúl Santiago “Learning Analytics. La narración del aprendizaje a través de los datos.”.

¿Y cómo se une esto con los MOOCs? Pues fácilmente se puede entender que los MOOCs son una herramienta de aprendizaje que recogen una gran cantidad de información acerca de sus usuarios (perfil, contribuciones, datos de registro, interacciones del usuario con el sistema...), por lo que el poder analizar toda esa información puede mejorar el aprendizaje desde tres puntos de vista: alumno, profesor y administrador.

Y esto seguirá en aumento con el paso de los años, pues cada vez se almacenará más y más información sobre los usuarios. En los MOOCs se habla de la versión 4.0, donde el aprendizaje sigue una estructura de “muchos para uno”, es decir, una persona puede aprender de muchas personas gracias al intercambio de ideas en las redes, todo ello generando una información que puede ser analizada.

En nuestro caso particular, la información que podemos sacar nos puede servir para mejorar el aprendizaje en los distintos aspectos que hemos enumerado anteriormente. Y con su visualización se puede analizar más fácilmente. Desde lo más básico, como ver el interés de los usuarios por un curso y corregir posibles deficiencias

en el mismo, hasta algo más complejo como realizar un seguimiento de todos los alumnos y analizar su rendimiento, o posibles fluctuaciones de éste, y por lo tanto, poder corregir actitudes de los alumnos.

Otros aspectos que se podrían analizar son: el rendimiento de los usuarios en relación con el tiempo de conexión en la plataforma (midiendo el nivel de productividad o viendo posibles problemas de aprendizaje de algún alumno), prever el comportamiento según la fecha o la hora del día que sea, o hasta poder llegar a analizar si ha habido algún evento inusual que le haya pasado al usuario, analizando su patrón de comportamiento.

Resumiendo, con los MOOCs y las Learning Analytics, se podrían analizar miles de datos de todos los usuarios para utilizarlos en una mejora del aprendizaje utilizando diferentes herramientas.

CAPÍTULO 5

DISEÑO E IMPLEMENTACIÓN DE LA SOLUCIÓN

5. Diseño e implementación de la solución

En este capítulo se detallará todo el trabajo realizado para alcanzar el objetivo del TFG.

5.1 Creación de datos

El primer paso a realizar, una vez tenemos todo instalado y funcionando, es crear datos en la aplicación para poder generar las visualizaciones.

Para poder empezar, primero lanzaremos tres servidores desde dentro de la plataforma de edX para que funcionen todos los componentes que nos hacen falta: LMS, Open edX Studio y Discussion Forums. Cada uno de ellos se lanza y se accede de manera diferente, y cada uno de ellos en diferentes terminales.

	<u>LMS</u>	<u>Studio</u>	<u>Forums</u>
Cómo lanzar	<code>~\$ sudo -Hu edxapp bash</code> <code>~\$ paver devstack lms</code>	<code>~\$ sudo -Hu edxapp bash</code> <code>~\$ paver devstack studio</code>	<code>~\$ sudo -Hu forum bash</code> <code>~\$ ruby app.rb -p 18080</code>
Cómo acceder	localhost:8000	localhost:8001	Pestaña foro de LMS

Tabla 1 - Funcionamiento de los diferentes componentes

Con los tres servidores corriendo, podemos acceder a los puertos 8000 y 8001 de nuestra máquina local, donde existe la plataforma de edX localmente para poder trabajar. En el puerto 8001 podremos crear cursos y modificarlos, mientras que en el 8000 podremos ver la interfaz normal de edX, donde podremos acceder a los cursos, realizarlos y comentar en los foros cualquier cosa.

El siguiente paso que realizaremos será crear los usuarios necesarios para generar datos y para poder acceder a la plataforma. Al ser una plataforma local no nos valen los usuarios reales creados en la web de edX, tenemos que crearlos en nuestro sistema y validarlos con el típico correo electrónico de confirmación, el cuál nos aparecerá en el terminal al crear el usuario.

Para nuestro estudio, crearemos tres usuarios, dos estudiantes y un administrador, el cual se encargará de crear el curso y validar algunas cosas exclusivas por tener ese cargo. En el sistema, por defecto, vienen cuatro usuarios ya creados que podremos utilizar si lo necesitamos: `honor@example.com`, `audit@example.com`, `verified@example.com` y `staff@example.com`.

A continuación, crearemos el curso con el que interactuaremos entre los distintos usuarios desde el componente de Studio. Para crear el curso existen guías diversas, e incluso, algún curso en línea en esta plataforma. También existe un curso por defecto añadido en la plataforma a modo de ejemplo. El curso que he creado contiene varios temas con diferentes partes, cada uno con un sistema diferente: video, checkbox, respuesta libre, opinión... Una vez está configurado el curso, hay que liberarlo para que sea visible y accesible para todos los usuarios.

Por último, accederemos con los tres usuarios de manera aleatoria al curso e iremos realizando el curso con los tres, generando comentarios y haciendo las diferentes tareas. Cualquier acción que haga cualquier usuario quedará registrada en un registro de seguimiento (tracking log), lo que nos será de gran ayuda para diversas consultas.

5.2 Consultas en MySQL

Para poder acceder a la base de datos de MySQL, deberemos arrancar la aplicación, y desde el directorio principal, introducir el comando “sudo mysql”. Esta base de datos de edX contiene datos de los usuarios en entornos de producción, es decir, información de los usuarios en su interacción con la plataforma, lo que nos valdrá para obtener información sobre inicios de sesión, tiempos de uso en la plataforma o preferencias de los usuarios, entre otras cosas. Todo ello se incluye en versiones de MySQL superiores a 5.1 con motor de almacenamiento InnoDB.

Dentro de este MySQL existen 7 bases de datos con diversa información, y dentro de ellas existen cientos de tablas diferentes. Al ser un entorno abierto y cambiante el de edX, varias de esas tablas están en desuso o obsoletas, por lo que se encuentran vacías. También se pueden encontrar nuevas tablas en cada nueva edición de la aplicación como consecuencia de su evolución constante. Casi todas las tablas se crean siguiendo las mismas convenciones (UTF-8 para archivos de salida, tiempos en UTC, \r para retornos de carro, \n para nuevas líneas, ...), pero existen excepciones.

```
vagrant@precise64:~$ sudo mysql
Welcome to the MySQL monitor.  Commands end with ; or \g.
Your MySQL connection id is 2
Server version: 5.6.35 MySQL Community Server (GPL)

Copyright (c) 2000, 2016, Oracle and/or its affiliates. All rights reserved.

Oracle is a registered trademark of Oracle Corporation and/or its
affiliates. Other names may be trademarks of their respective
owners.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

mysql> show databases;
+-----+
| Database |
+-----+
| information_schema |
| ecommerce |
| edxapp |
| edxapp_csmh |
| mysql |
| performance_schema |
| programs |
+-----+
7 rows in set (0.00 sec)

mysql> 
```

Figura 4 - Conexión a MySQL y lista de bases de datos

Después de ver y analizar todas las bases de datos, para nuestro estudio particular utilizaremos únicamente la de edxapp. Ésta contiene 305 tablas de variado contenido: registro de seguimiento, datos de usuarios, datos de los alumnos de cada curso, información de los cursos, certificados...

90	2017-04-20 09:52:10.147375	DaniG	10.0.2.2 server /notification_prefs/status/
91	2017-04-20 10:40:38.372950	DaniG	10.0.2.2 server /courses/course-v1:UC3M+ST_1+2016_4/info
92	2017-04-20 10:55:55.215556	DaniG	10.0.2.2 server /logout
93	2017-04-20 10:55:55.278776		10.0.2.2 server /
94	2017-04-20 10:57:28.630056		10.0.2.2 server /user_api/v1/account/login_session/
95	2017-04-20 10:57:29.118017	DaniG	10.0.2.2 server /dashboard
96	2017-04-20 11:01:20.330862	DaniG	10.0.2.2 server /courses/course-v1:UC3M+ST_1+2016_4/info

Figura 5 - Ejemplo de datos registrados en el tracking log

De todas las posibles consultas realizaremos estas:

- **Tiempo en los cursos:** se calculará el tiempo total que ha estado cada usuario, desde que inicia sesión hasta que se desconecta, todas las veces que se haya conectado desde que se creó el curso. Estos datos se sacan de la tabla track_trackinglog.

```
USE edxapp
CREATE TABLE consulta (usuario varchar(32), dtcreated datetime(6), event_type varchar(512)) SELECT
CASE
  WHEN event_type = '/user_api/v1/account/login_session/'
  THEN auth_user.username
  ELSE track_trackinglog.username
END AS usuario, track_trackinglog.dtcreated, track_trackinglog.event_type
FROM track_trackinglog
LEFT JOIN auth_user ON auth_user.email = substr(event, instr(event, "email": [""] + 11, instr(event, '"') - instr(event, "email": [""] - 11)
WHERE event_type IN ('/logout', '/user_api/v1/account/login_session/');

USE edxapp
SELECT usuario,
sec_to_time(sum((unix_timestamp(dtcreated) - unix_timestamp(( select max(dtcreated) FROM consulta
WHERE event_type='/user_api/v1/account/login_session/' and alias1.usuario=usuario and alias1.dtcreated >= dtcreated ))))) AS tiempo_con
FROM consulta AS alias1
WHERE event_type = '/logout'
GROUP BY usuario
ORDER BY usuario;

USE edxapp
DROP TABLE consulta;
```

Figura 6 - Código tiempo en los cursos

- **Número de visitas:** número de veces que ha iniciado sesión cada usuario. Al igual que la anterior consulta, los datos se sacan de la tabla track_trackinglog.

```
USE edxapp
CREATE TABLE consulta (usuario varchar(32), dtcreated datetime(6), event_type varchar(512)) SELECT
CASE
    WHEN event_type = '/user_api/v1/account/login_session/'
    THEN auth_user.username
    ELSE track_trackinglog.username
END AS usuario, track_trackinglog.dtcreated, track_trackinglog.event_type
FROM track_trackinglog
LEFT JOIN auth_user ON auth_user.email = substr(event, instr(event, 'email': ['') + 11, instr(event, '[') - instr(event, 'email': ['') - 11)
WHERE event_type IN ('/logout', '/user_api/v1/account/login_session/');

USE edxapp
SELECT usuario, dtcreated
FROM consulta
WHERE event_type='/user_api/v1/account/login_session/'
ORDER BY dtcreated;

USE edxapp
DROP TABLE consulta;
```

Figura 7 - Código número de visitas

- **Conexiones en el tiempo:** distribución de conexiones de cada usuario desde el principio. En esta también se utiliza la tabla track_trackinglog.

```
USE edxapp
CREATE TABLE consulta (usuario varchar(32), dtcreated datetime(6), event_type varchar(512)) SELECT
CASE
    WHEN event_type = '/user_api/v1/account/login_session/'
    THEN auth_user.username
    ELSE track_trackinglog.username
END AS usuario, track_trackinglog.dtcreated, track_trackinglog.event_type
FROM track_trackinglog
LEFT JOIN auth_user ON auth_user.email = substr(event, instr(event, 'email': ['') + 11, instr(event, '[') - instr(event, 'email': ['') - 11)
WHERE event_type IN ('/logout', '/user_api/v1/account/login_session/');

USE edxapp
SELECT usuario, dtcreated
FROM consulta
WHERE event_type='/user_api/v1/account/login_session/'
ORDER BY dtcreated;

USE edxapp
DROP TABLE consulta;
```

Figura 8 - Código conexiones en el tiempo

- **Número de alumnos por curso:** cantidad de usuarios apuntados a cada curso. En este caso, la tabla que utilizaremos es student_courseenrollment.

```
USE edxapp
SELECT course_id,
COUNT(is_active) AS alumnos_activos
FROM edxapp.student_courseenrollment
GROUP BY course_id;
```

Figura 9 - Código número de alumnos por curso

- **Última conexión:** consultar la última vez que cada usuario se conectó a la plataforma. La tabla utilizada es auth_user.

```
USE edxapp
SELECT username, last_login
FROM edxapp.auth_user;
```

Figura 10 - Código última conexión

- **Comentarios totales distribuidos en el tiempo:** fechas de creación de cada comentario y el usuario que lo creó. Se utiliza la tabla track_trackinglog.

```
USE edxapp
SELECT username, dtcreated
FROM edxapp.track_trackinglog
WHERE event_type like '%reply%'
OR event_type like '%/create%'
ORDER BY dtcreated;
```

Figura 11 - Código comentarios totales distribuidos en el tiempo

- **Hilos iniciados distribuidos en el tiempo:** en este caso sólo se seleccionarán los comentarios que inician un nuevo hilo en el foro. Se utiliza la tabla track_trackinglog.

```
USE edxapp
SELECT username, dtcreated
FROM edxapp.track_trackinglog
WHERE event_type like '%/create%'
ORDER BY dtcreated;
```

Figura 12 - Código hilos iniciados distribuidos en el tiempo

- **Respuestas distribuidas en el tiempo:** únicamente se seleccionan las respuestas generadas en los hilos del foro. Se utiliza la tabla track_trackinglog.

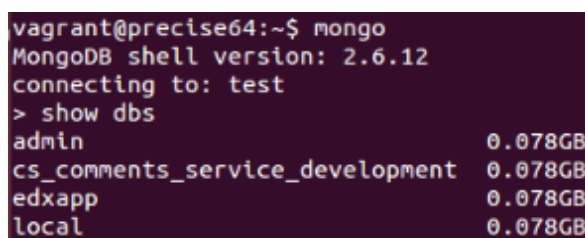
```
USE edxapp
SELECT username, dtcreated
FROM edxapp.track_trackinglog
WHERE event_type like '%reply%'
ORDER BY dtcreated;
```

Figura 13 - Código respuestas distribuidas en el tiempo

5.3 Consultas en MongoDB

Para poder acceder a la base de datos de MongoDB, deberemos arrancar la aplicación y desde el directorio principal introducir el comando “mongo”. Esta base de datos de edX contiene los datos pedagógicos de la plataforma, es decir, toda la información creada en los cursos (temario, secciones, ejercicios...) y toda la información de los foros (comentarios, “likes”, hilos...), lo que nos valdrá para realizar consultas sobre cualquier cuestión posible sobre los foros o sobre los contenidos de los cursos.

Dentro de este MongoDB existen 4 bases de datos con diversa información, y dentro de ellas hay 15 documentos JSON en total, donde se guardan todos los datos. Al igual que ocurre en el caso de MySQL, el entorno de edX va cambiando con cada nueva edición, y existen campos de los documentos que ya no se utilizan.



```
vagrant@precise64:~$ mongo
MongoDB shell version: 2.6.12
connecting to: test
> show dbs
admin 0.078GB
cs_comments_service_development 0.078GB
edxapp 0.078GB
local 0.078GB
```

Figura 14 - Conexión a MongoDB y lista de bases de datos

Para nuestro proyecto, después de analizar todas, utilizaremos la base de datos `cs_comments_service_development`, y más concretamente, el documento `contents`, que almacena toda la información relacionada con los foros.

El documento `contents` contiene una entrada por cada comentario que se ha hecho en los foros de todos los cursos. Cada una de las entradas contiene toda la información necesaria para nuestro objetivo. Por ejemplo, quién creó el comentario, cantidad de votos positivos que tiene (los votos negativos están en desuso en la aplicación), a qué curso pertenece, a qué hilo pertenece... e incluso el comentario que se puso aparece entero y sin cifrar.

```
{
  "_id" : ObjectId("58a0ab8a56c02c33f8000003"),
  "votes" : {
    "up" : [
      "7"
    ],
    "down" : [ ],
    "up_count" : 1,
    "down_count" : 0,
    "count" : 1,
    "point" : 1
  },
  "visible" : true,
  "abuse_flaggers" : [ ],
  "historical_abuse_flaggers" : [ ],
  "_type" : "CommentThread",
  "thread_type" : "discussion",
  "context" : "course",
  "comment_count" : 1,
  "at_position_list" : [ ],
  "title" : "Curso en general",
  "body" : "¿Como veis el curso?",
  "course_id" : "course-v1:UC3M+ST_1+2016_4",
  "commentable_id" : "course",
  "anonymous" : false,
  "anonymous_to_peers" : false,
  "closed" : false,
  "author_id" : "6",
  "author_username" : "dvd1pzmrz88",
  "updated_at" : ISODate("2017-02-20T07:10:05.039Z"),
  "created_at" : ISODate("2017-02-12T18:38:02.368Z"),
  "last_activity_at" : ISODate("2017-02-20T07:10:05.038Z")
}
```

Figura 15 - Ejemplo de entrada en el documento contents

Las consultas que realizaremos son:

- **Cantidad de hilos que inicia un usuario:** número de hilos que abre cada usuario viendo el tipo de comentario que es. En este caso, es de tipo “CommentThread”.

```
db.contents.aggregate(
  [
    {$match:{_type:"CommentThread"}},
    {$group:{
      _id:{id:$author_id, author:$author_username},
      count:{$sum:1}
    }}
  ]
)
```

Figura 16 - Código cantidad de hilos que inicia un usuario

- **Cantidad de respuestas que hace un usuario:** número de comentarios que realiza un usuario contestando a otros usuarios. Para saberlo tendremos que fijarnos en que el tipo de comentario sea “Comment”.

```
db.contents.aggregate(  
  [  
    {$match: {_type: "Comment"}},  
    {$group: {  
      _id: {id: $author_id, author: $author_username},  
      count: {$sum: 1}  
    }}  
  ]  
)
```

Figura 17 - Código cantidad de respuestas que hace un usuario

- **Cantidad total de comentarios:** número total de comentarios de cada usuario.

```
db.contents.aggregate(  
  [  
    {$group: {  
      _id: {"author_id", author: $author_username},  
      count: {$sum: 1}  
    }}  
  ]  
) .pretty()
```

Figura 18 - Código cantidad total de comentarios

- **Perfil del usuario:** ver el perfil de cada usuario en relación a su actividad en los foros, es decir, si es preguntador o respondedor.

```
db.contents.aggregate(  
  [  
    {$group: {  
      _id: {id: "$author_id", type: "$_type"},  
      count: {$sum: 1}  
    }}  
  ]  
)
```

Figura 19 - Código perfil de usuario

- **Media de comentarios que genera cada usuario:** calcula la media de los comentarios que genera cada usuario al iniciar hilos. Permite saber si un usuario suele hacer preguntas que generan discusión o no.

```
db.contents.aggregate(  
  [  
    {$match:{_type:"CommentThread"},  
    {$group:{  
      _id:{id:$author_id, author:$author_username},  
      averageQuantity:{$avg:"$comment_count"}  
    }}  
  ]  
)
```

Figura 20 - Código media de comentarios que genera cada usuario

- **Calidad de los comentarios:** cantidad de positivos que tiene cada usuario gracias a sus comentarios en el foro.

```
db.contents.aggregate(  
  [  
    {$group:{  
      _id:{id:$author_id, author:$author_username},  
      count:{$sum:"$votes.up_count"}  
    }}  
  ]  
)
```

Figura 21 - Código calidad de los comentarios

- **Calidad de las preguntas:** cantidad de positivos totales que tiene cada usuario en sus preguntas.

```
db.contents.aggregate(  
  [  
    {$match:{_type:"CommentThread"}},  
    {$group:{  
      _id:{id:$author_id, author:$author_username},  
      count:{$sum:"$votes.up_count"}  
    }}  
  ]  
)
```

Figura 22 - Código calidad de las preguntas

- **Calidad de las respuestas:** cantidad de positivos totales que tiene cada usuario por sus respuestas en el foro.

```
db.contents.aggregate(
  [
    {$match: {_type: "Comment"}},
    {$group: {
      _id: {id: $author_id, author: $author_username},
      count: {$sum: "votes.up_count"}
    }}
  ]
)
```

Figura 23 - Código calidad de las respuestas

- **Calidad media:** media de todos los positivos de cada usuario entre todos los comentarios que ha realizado. Sirve para saber si un usuario realiza comentarios de calidad o no.

```
db.contents.aggregate(
  [
    {$group: {
      _id: {id: $author_id, author: $author_username},
      count: {$avg: "votes.up_count"}
    }}
  ]
)
```

Figura 24 - Código calidad media

Todas estas consultas se realizan sobre todos los cursos que haya en la aplicación. Para obtener la información de un curso concreto hay que añadir una línea de código, y hay que saber el identificador del curso. En el ejemplo a continuación, se utiliza el identificador del curso que se creó anteriormente.

```
{$match: { course_id: "course-v1:UC3M+ST_1+2016_4" }},
```

Figura 25 - Línea de código para elegir curso

5.4 Visualizaciones con D3.js

Para poder ver las visualizaciones, primero tenemos que instalar un par de herramientas que nos ayuden: servidor HTTP Apache y php. Todo ello nos servirá para poder ver, en cualquier buscador web, todas las visualizaciones que hagamos.

El funcionamiento de todo es muy sencillo. Una vez instalemos las dos herramientas, podremos incluir en el directorio de html (/var/www/html/) los documentos php y html necesarios para sacar nuestras visualizaciones. Después nos conectamos, mediante cualquier explorador, a localhost donde aparecerán todos los documentos incluidos en ese directorio, y pinchando en cada uno veremos las visualizaciones (en el caso de los documentos html) o los resultados de cada consulta en las BBDD (en caso de los documentos php).

Index of /

	<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
	calidadcomentariosmongodb.html	2017-06-06 19:33	2.8K	
	calidadcomentariosmongodb.php	2017-06-06 19:24	497	
	calidadmediamongodb.html	2017-06-06 19:35	2.2K	
	calidadmediamongodb.php	2017-06-06 19:24	497	
	calidadpreguntasmongodb.html	2017-06-06 19:36	3.0K	
	calidadpreguntasmongodb.php	2017-06-06 19:24	497	
	calidadrespuestasmongodb.html	2017-06-06 19:35	2.2K	
	calidadrespuestasmongodb.php	2017-06-06 19:24	497	
	comentariosalestiemposmysql.html	2017-06-06 19:33	2.8K	
	comentariosalestiemposmysql.php	2017-06-06 19:13	609	
	conexionestiemposmysql.html	2017-06-06 19:37	2.0K	
	conexionestiemposmysql.php	2017-06-06 19:12	609	
	hiloiniciadosusuariosmysql.html	2017-06-06 19:35	2.2K	
	hiloiniciadosusuariosmysql.php	2017-06-06 19:13	609	
	mediacomentariosusuariosmongodb.html	2017-06-06 19:34	2.8K	
	mediacomentariosusuariosmongodb.php	2017-06-06 19:21	497	
	numeroalumnoscursomysql.html	2017-06-06 19:37	3.0K	
	numeroalumnoscursomysql.php	2017-06-06 19:14	609	
	numeroconexionesusuariosmysql.html	2017-06-06 19:35	2.2K	
	numeroconexionesusuariosmysql.php	2017-06-06 19:16	1.1K	
	numeropostiniciomongodb.html	2017-06-06 19:37	2.0K	
	numeropostiniciomongodb.php	2017-06-06 19:19	497	
	numeropostrespuestasmongodb.html	2017-06-06 19:34	2.8K	
	numeropostrespuestasmongodb.php	2017-06-06 19:20	497	
	numeroposttotalesmongodb.html	2017-06-06 19:36	3.0K	
	numeroposttotalesmongodb.php	2017-06-06 19:20	497	
	perfilusuariosmongodb.html	2017-06-06 19:37	2.0K	
	perfilusuariosmongodb.php	2017-06-06 19:21	497	
	phpmyadmin/	2013-12-04 18:44	-	
	respuestashilosusuariosmysql.html	2017-06-06 19:36	3.0K	
	respuestashilosusuariosmysql.php	2017-06-06 19:15	1.1K	
	tiempototalmysql.html	2017-06-06 19:35	2.2K	
	tiempototalmysql.php	2017-06-06 19:15	1.1K	
	ultimaconexionmysql.html	2017-06-06 19:34	2.8K	
	ultimaconexionmysql.php	2017-06-06 19:13	609	

Apache/2.4.7 (Ubuntu) Server at localhost Port 80

Figura 26 - Documentos que aparecen en localhost

Para crear los documentos php utilizaremos 2 versiones básicas, una para cada tipo de BBDD. Cada documento se guarda con un nombre diferente, ya que los documentos html los llamarán más tarde y cada uno de ellos es distinto.

En el caso de las consultas de MySQL, el código utilizado realiza primero una conexión a la base de datos a la que queremos conectarnos. Para ello, rellenaremos una serie de campos con la información básica de la base de datos: usuario, contraseña, "host" y nombre de la base de datos. Después realizamos la consulta que queremos hacer y devolvemos la información en un archivo json, que más tarde utilizará el documento html.

```
<?php
    $username = "*****";
    $password = "*****";
    $host = "*****";
    $database="*****";

    $server = mysql_connect($host, $username, $password);
    $connection = mysql_select_db($database, $server);

    $myquery = "Consultas aqui";
    $query = mysql_query($myquery);

    if ( ! $query ) {
        echo mysql_error();
        die;
    }

    $data = array();

    for ($x = 0; $x < mysql_num_rows($query); $x++) {
        $data[] = mysql_fetch_assoc($query);
    }

    echo json_encode($data);
    mysql_close($server);
?>
```

Figura 27 - Código php consultas MySQL

Para algunos casos se tendrá que añadir dos líneas antes de cerrar la conexión con el servidor, ya que en algunas consultas se crea una tabla para poder realizar la consulta sin fallos, y para mantener la base de datos en su estado original, hay que eliminar esa tabla. Esas líneas son las de \$myquery y \$query, donde la consulta que se insertará será la necesaria para eliminar la tabla.

Para el caso de las consultas de MongoDB, el código utilizado es muy parecido al anterior, sobre todo en su funcionamiento. Primero se conecta, luego se selecciona la base de datos y después se selecciona la colección que queremos. A continuación, incluimos la consulta que necesitamos y se guardan los datos.

```
<?php

$salted = "${username}:mongo:${password}";
$hash = md5($salted);

$nonce = $db->command(array("getnonce" => 1));

$saltedHash = md5($nonce["nonce"]."${username}${hash}");

$db->command(array("authenticate" => 1,
    "user" => $username,
    "nonce" => $nonce["nonce"],
    "key" => $saltedHash
));

$m = new MongoClient();
$db = $m->selectDB('*****');
$colección = new MongoClient($db, 'cs_comments_service_development');

$rangeQuery = array('consulta');

$cursor = $colección->find($rangeQuery);
foreach ($cursor as $doc) {
    var_dump($doc);
}

$db->command(array("logout" => 1));

?>
```

Figura 28 - Código php consultas MongoDB

El siguiente paso es hacer todos los documentos html para cada consulta, y cada uno estará relacionado con cada documento php hecho antes. Html da una libertad enorme para realizar modificaciones en todas las visualizaciones, y unido a D3.js hace que haya miles de opciones posibles para ver los resultados.

En estos documentos hay dos partes de código esenciales para obtener el resultado que queremos: una parte donde se importan las librerías de D3.js y otra parte donde se obtienen los datos de las consultas llamando al documento php correspondiente. Ambas se pueden ver a continuación:

```
<script src="https://d3js.org/d3.v4.min.js"></script>
```

Figura 29 - Línea código llamada librería D3.js

```
d3.json("documento.php", function(error, data) {  
  data.forEach(function(d) {  
    d.date = parseDate(d.date);  
    d.close = +d.close;  
  });  
});
```

Figura 30 - Ejemplo de código de llamada a los documentos php

CAPÍTULO 6

VISUALIZACIONES

6. Visualizaciones

En este capítulo se explicarán las visualizaciones realizadas.

6.1 Visualizaciones

Una vez tenemos todo el proceso claro, el siguiente paso es escribir las líneas de código necesarias en html para que se visualicen correctamente todas las consultas. Para no hacer el proyecto demasiado extenso, no se incluirá el código completo de todos los html creados, ya que se necesitarían muchas páginas y se sobrepasaría el número de páginas indicadas en los requisitos del TFG. Eso sí, se incluirán algunos ejemplos del código y varias visualizaciones de las posibles que se pueden hacer:

- Primer ejemplo: **calidad de los comentarios.**

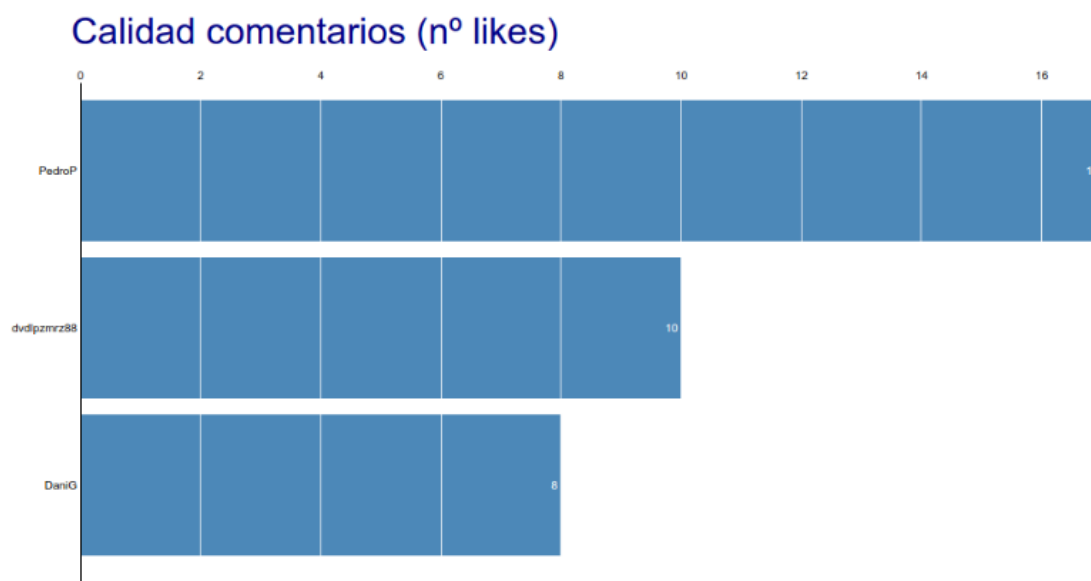


Figura 31 – Visualización calidad de los comentarios

```

<!DOCTYPE html>
<meta charset="utf-8">

<head>
  <title>HTML Reference</title>
</head>

<style>

body {
  font: 10px sans-serif;
}

.bar rect {
  fill: steelblue;
}

.bar text.dtccreated {
  fill: white;
}

.axis {
  shape-rendering: crispEdges;
}

.axis path {
  fill: none;
}

.x.axis line {
  stroke: #fff;
  stroke-opacity: .8;
}

.y.axis path {
  stroke: black;
}

</style>
<body>
<script src="//d3js.org/d3.v3.min.js"></script>
<div style="position:absolute; top:5px;left:60px;">
<FONT SIZE=6 COLOR=NAVY> Calidad comentarios (nº likes)</FONT>
</div>
<script>

var m = [60, 10, 5, 60],
    w = 960 - m[1] - m[3],
    h = 630 - m[0] - m[2];

var format = d3.format(",.0f");

var x = d3.scale.linear().range([0, w]),
    y = d3.scale.ordinal().rangeRoundBands([0, h], .1);

var xAxis = d3.svg.axis().scale(x).orient("top").tickSize(-h),
    yAxis = d3.svg.axis().scale(y).orient("left").tickSize(0);

var svg = d3.select("body").append("svg")
  .attr("width", w + m[1] + m[3])
  .attr("height", h + m[0] + m[2])
  .append("g")
  .attr("transform", "translate(" + m[3] + "," + m[0] + ")");

d3.json("calidadcomentariosmongodb.php", function(error, data) {
  if (error) throw error;

```



```
svg.append("g")
    .attr("class", "y axis")
    .call(yAxis);
});
</script>
```

Figura 32 – Código html calidad de los comentarios

En este ejemplo podemos ver un html sencillo. En la primera parte se codifica el estilo de la visualización, donde se elige el color de las barras, el tipo de letra, etc. En la segunda parte se definen las peculiaridades de esta visualización: barras horizontales, posición y tamaño de la gráfica...

- Ejemplo 2: **perfil de usuario.**



Figura 33 – Visualización perfil de usuario

```

<!DOCTYPE html>
<style>

.axis .domain {
  display: none;
}

</style>
<svg width="960" height="600"></svg>
<script src="https://d3js.org/d3.v4.min.js"></script>
<div style="position:absolute; top:5px;left:60px;">
<FONT SIZE=6 COLOR=NAVY> Perfil de usuario (preguntador o respondedor)</FONT>
</div>
<script>

var svg = d3.select("svg"),
    margin = {top: 60, right: 20, bottom: 30, left: 40},
    width = +svg.attr("width") - margin.left - margin.right,
    height = +svg.attr("height") - margin.top - margin.bottom,
    g = svg.append("g").attr("transform", "translate(" + margin.left + "," + margin.top + ")");

var x0 = d3.scaleBand()
    .rangeRound([0, width])
    .paddingInner(0.1);

var x1 = d3.scaleBand()
    .padding(0.05);

var y = d3.scaleLinear()
    .rangeRound([height, 0]);

var z = d3.scaleOrdinal()
    .range(["#98abc5", "#ff8c00"]);

d3.json("perfilusuariomongodb.php", function(d, i, columns) {
  for (var i = 1, n = columns.length; i < n; ++i) d[columns[i]] = +d[columns[i]];

  return d;
}, function(error, data) {
  if (error) throw error;

  var keys = data.columns.slice(1);

  x0.domain(data.map(function(d) { return d.Usuario; }));
  x1.domain(keys).rangeRound([0, x0.bandwidth()]);
  y.domain([0, d3.max(data, function(d) { return d3.max(keys, function(key) { return d[key]; }); })]).nice();

  g.append("g")
    .selectAll("g")
    .data(data)
    .enter().append("g")
      .attr("transform", function(d) { return "translate(" + x0(d.Usuario) + ",0)"; })
    .selectAll("rect")
    .data(function(d) { return keys.map(function(key) { return {key: key, value: d[key]}; }); })
    .enter().append("rect")
      .attr("x", function(d) { return x1(d.key); })
      .attr("y", function(d) { return y(d.value); })
      .attr("width", x1.bandwidth())
      .attr("height", function(d) { return height - y(d.value); })
      .attr("fill", function(d) { return z(d.key); });

  g.append("g")
    .attr("class", "axis")
    .attr("transform", "translate(0," + height + ")")
    .call(d3.axisBottom(x0));

```

```

g.append("g")
  .attr("class", "axis")
  .call(d3.axisLeft(y).ticks(null, "s"))
  .append("text")
  .attr("x", 2)
  .attr("y", y(y.ticks().pop()) + 0.5)
  .attr("dy", "0.32em")
  .attr("fill", "#000")
  .attr("font-weight", "bold")
  .attr("text-anchor", "start");

var legend = g.append("g")
  .attr("font-family", "sans-serif")
  .attr("font-size", 10)
  .attr("text-anchor", "end")
  .selectAll("g")
  .data(keys.slice().reverse())
  .enter().append("g")
  .attr("transform", function(d, i) { return "translate(0," + i * 20 + ")"; });

legend.append("rect")
  .attr("x", width - 19)
  .attr("width", 19)
  .attr("height", 19)
  .attr("fill", z);

legend.append("text")
  .attr("x", width - 24)
  .attr("y", 9.5)
  .attr("dy", "0.32em")
  .text(function(d) { return d; });
});
</script>

```

Figura 34 – Código html perfil de usuario

En este ejemplo, no definimos el estilo en primer lugar por separado del cuerpo del código, sino que se define todo junto en un script. Las particularidades de esta visualización son las dos columnas por usuario y la información adicional en la esquina superior, esta última definida con la variable “legend”.

- Ejemplo 3: **número de conexiones por usuarios.**

Numero de conexiones

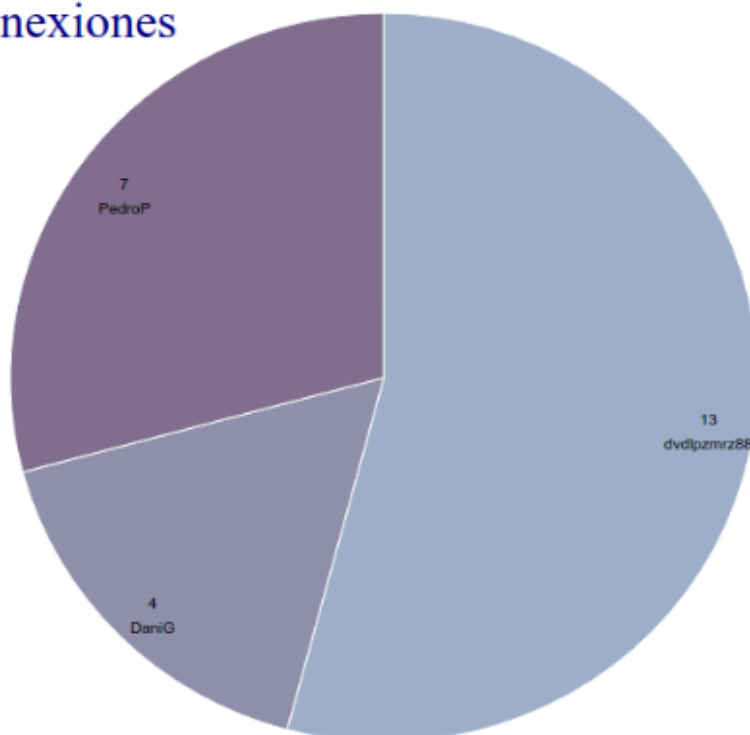


Figura 35 – Visualización número de conexiones por usuario

```
<!DOCTYPE html>
<meta charset="utf-8">
<style>

.arc text {
  font: 10px sans-serif;
  text-anchor: middle;
}

.arc path {
  stroke: #fff;
}

</style>
<svg width="960" height="500"></svg>
<script src="https://d3js.org/d3.v4.min.js"></script>
<div style="position:absolute; top:5px;left:60px;">
<FONT SIZE=6 COLOR=NAVY> Numero de conexiones</FONT>
</div>
<script>

var svg = d3.select("svg"),
    width = +svg.attr("width"),
    height = +svg.attr("height"),
    radius = Math.min(width, height) / 2,
    g = svg.append("g").attr("transform", "translate(" + width / 2 + "," + height / 2 + ")");

var color = d3.scaleOrdinal(["#98abc5", "#8a89a6", "#7b6888"]);

var pie = d3.pie()
  .sort(null)
  .value(function(d) { return d.numero_loggings; });

var path = d3.arc()
  .outerRadius(radius - 10)
  .innerRadius(0);
```

```
var label = d3.arc()  
  .outerRadius(radius - 40)  
  .innerRadius(radius - 40);  
  
d3.csv("numeroconexionesusuariomysql.csv", function(d) {  
  d.numero_loggings = +d.numero_loggings;  
  return d;  
}, function(error, data) {  
  if (error) throw error;  
  
  var arc = g.selectAll(".arc")  
    .data(pie(data))  
    .enter().append("g")  
    .attr("class", "arc");  
  
  arc.append("path")  
    .attr("d", path)  
    .attr("fill", function(d) { return color(d.data.username); });  
  
  arc.append("text")  
    .attr("transform", function(d) { return "translate(" + label.centroid(d) + ")"; })  
    .attr("dy", "0.35em")  
    .text(function(d) { return d.data.numero_loggings; });  
  
  arc.append("text")  
    .attr("transform", function(d) { return "translate(" + label.centroid(d) + ")"; })  
    .attr("dy", "1.95em")  
    .text(function(d) { return d.data.username; });  
});  
</script>
```

Figura 36 – Código html número de conexiones por usuario

En este caso, para poder realizar la circunferencia se llama a la función “pie()” y para poder incluir los datos en ella se utiliza la función “arc()”. Para incluir el texto con los usuarios y el número de conexiones de cada uno se utiliza el comando “append(“text”)” llamando a la variable arc.

- Ejemplo 4: **número de post totales.**

Numero de post totales

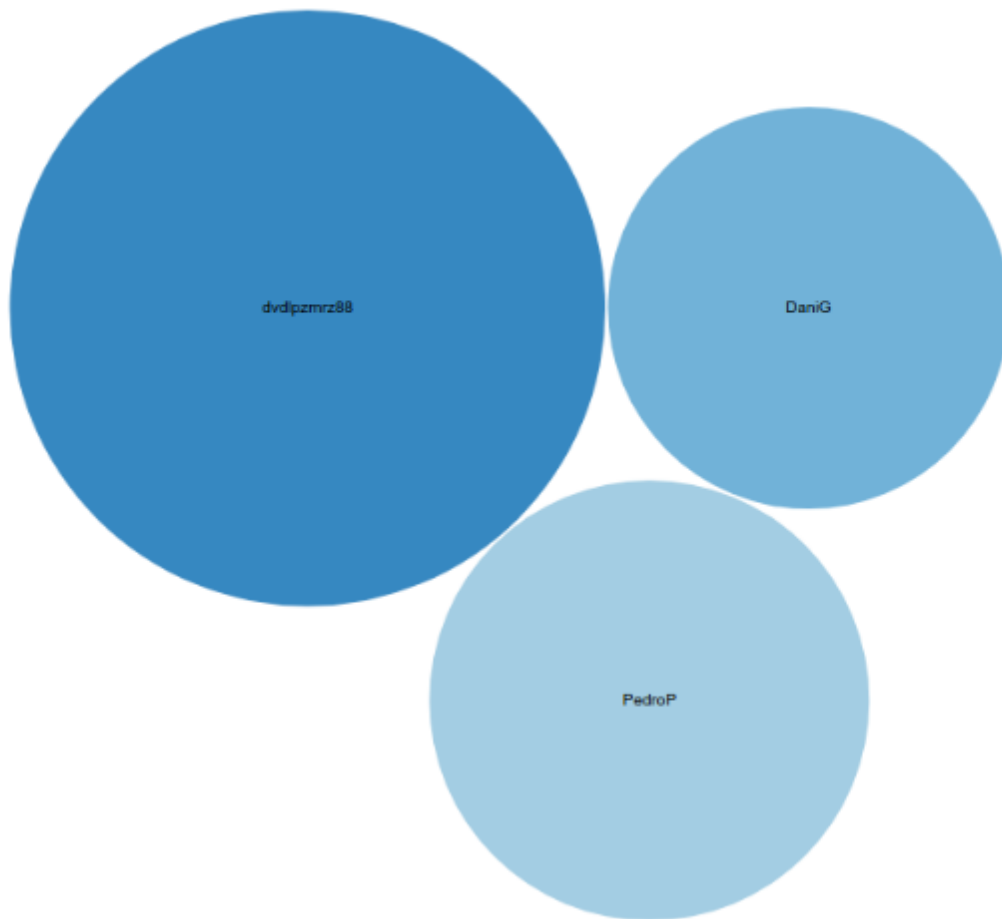


Figura 37 – Visualización número de post totales

Para esta ocasión, se utiliza una visualización con diferentes circunferencias o burbujas, en las que se miden los resultados dependiendo del tamaño de éstas. Para obtener la información más exacta, con dejar el cursor encima de la circunferencia, aparece un recuadro con la información del usuario y la cantidad del dato que queríamos.

```

<!DOCTYPE html>
<svg width="660" height="660" font-family="sans-serif" font-size="10" text-anchor="middle"></svg>
<script src="https://d3js.org/d3.v4.min.js"></script>
<div style="position:absolute; top:5px;left:60px;">
<FONT SIZE=10 COLOR=navy> Numero de post totales</FONT>
</div>
<script>

var svg = d3.select("svg"),
    width = +svg.attr("width"),
    height = +svg.attr("height");
|
var format = d3.format(",d");

var color = d3.scaleOrdinal(d3.schemeCategory20c);

var pack = d3.pack()
    .size([width, height])
    .padding(1.5);

d3.csv("burbujas.csv", function(d) {
    d.count = +d.count;
    if (d.count) return d;
}, function(error, classes) {
    if (error) throw error;

    var root = d3.hierarchy({children: classes})
        .sum(function(d) { return d.value; })
        .each(function(d) {
            if (id = d.data.id) {
                var id, i = id.lastIndexOf(".");
                d.id = id;
                d.package = id.slice(0, i);
                d.class = id.slice(i + 1);
            }
        });

    var node = svg.selectAll(".node")
        .data(pack(root).leaves())
        .enter().append("g")
        .attr("class", "node")
        .attr("transform", function(d) { return "translate(" + d.x + "," + d.y + ")"; });

    node.append("circle")
        .attr("id", function(d) { return d.id; })
        .attr("r", function(d) { return d.r; })
        .style("fill", function(d) { return color(d.package); });

    node.append("clipPath")
        .attr("id", function(d) { return "clip-" + d.id; })
        .append("use")
        .attr("xlink:href", function(d) { return "#" + d.id; });

    node.append("text")
        .attr("clip-path", function(d) { return "url(#clip-" + d.id + ")"; })
        .selectAll("tspan")
        .data(function(d) { return d.class.split(/(?=[A-Z][^A-Z])/g); })
        .enter().append("tspan")
        .attr("x", 0)
        .attr("y", function(d, i, nodes) { return 13 + (i - nodes.length / 2 - 0.5) * 10; })
        .text(function(d) { return d; });

    node.append("title")
        .text(function(d) { return d.id + "\n" + format(d.value); });
});
</script>

```

Figura 38 – Código html número de post totales

A continuación, se enseñarán algunos ejemplos de visualizaciones más, pero ahora sin el código fuente:

- Ejemplo 5: **conexiones en el tiempo**, donde el eje x es el tiempo, y el y la cantidad de conexiones.

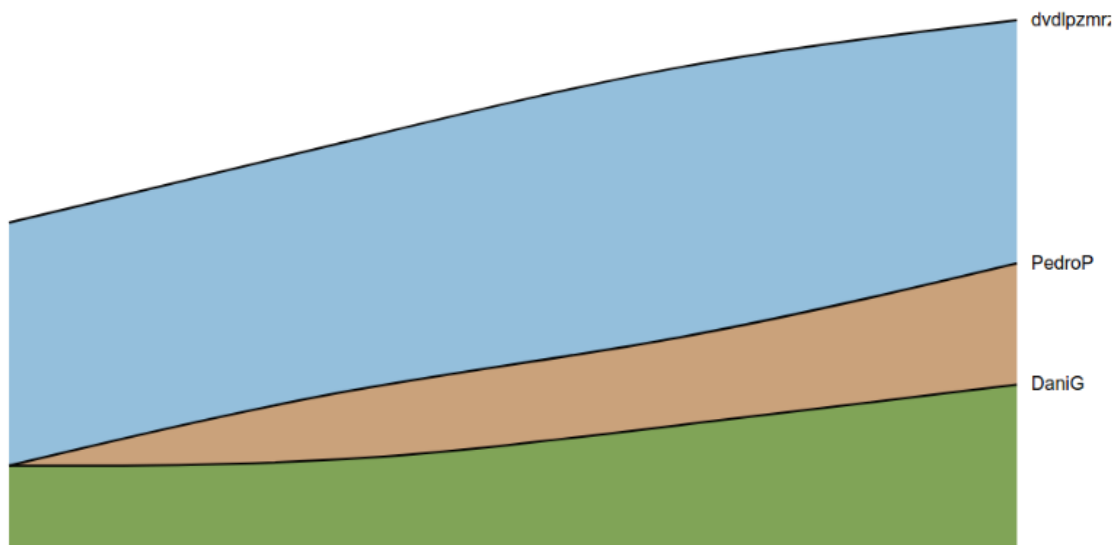


Figura 39 – Visualización de conexiones en el tiempo

- Ejemplo 6: **comentarios totales distribuidos en el tiempo**, donde el eje x es el tiempo y el eje y el número de comentarios.



Figura 40 – Visualización de comentarios totales distribuidos en el tiempo

- Ejemplo 7: **hilos iniciados distribuidos en el tiempo**, donde el eje x es el tiempo y el eje y el número de hilos iniciados por cada usuario.

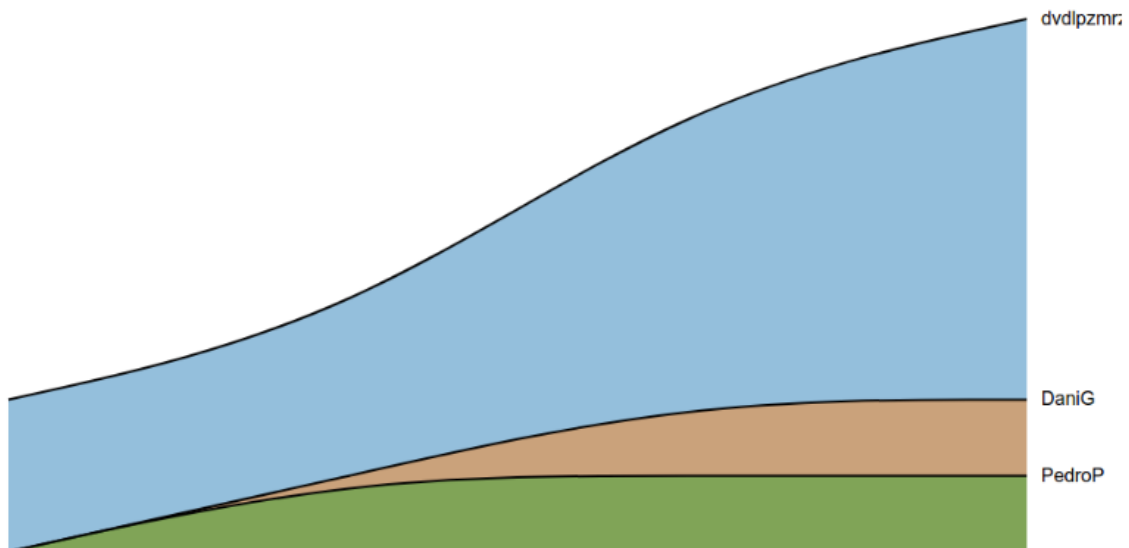


Figura 41 – Visualización de hilos iniciados distribuidos en el tiempo

- Ejemplo 8: **respuestas distribuidas en el tiempo**, donde el eje x son los meses de marzo a junio, y el eje y el número de respuestas de cada usuario.

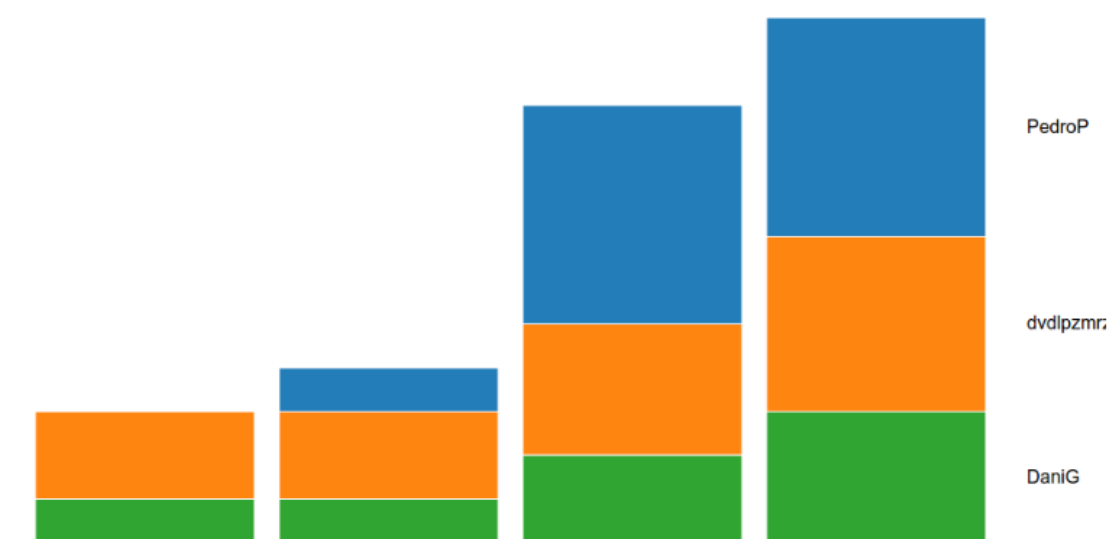


Figura 42 – Visualización de respuestas distribuidas en el tiempo

- Ejemplo 9: en este ejemplo se ha juntado la información de las consultas de **calidad de los comentarios** y se ha realizado una gráfica conjunta. Al poner el cursor encima de cada nombre conseguiríamos que sólo quedara marcada la línea de ese usuario.

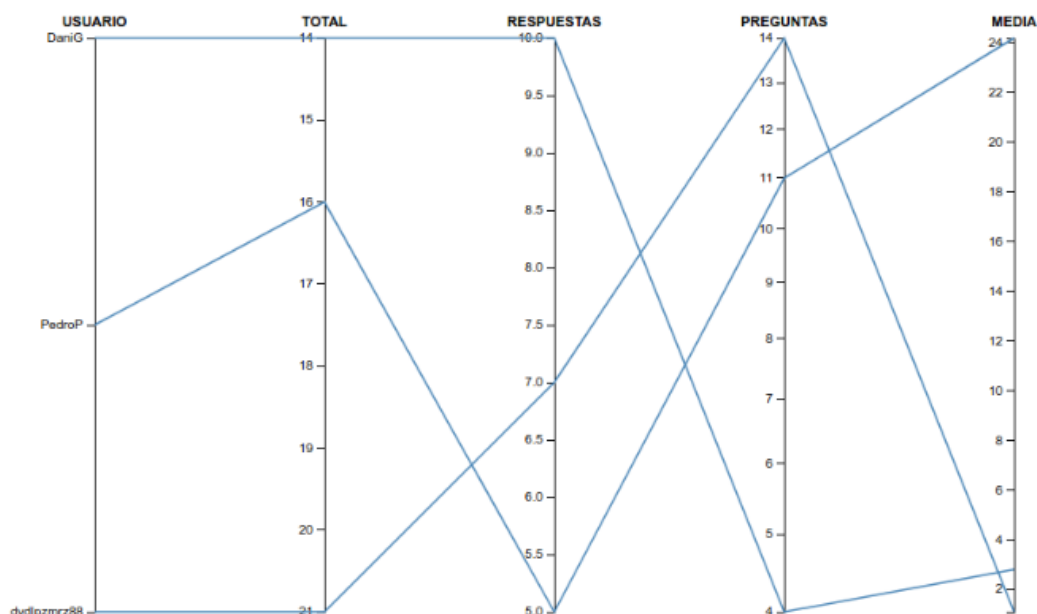


Figura 43 – Visualización calidad general de los comentarios.

Para esta memoria se han escogido las visualizaciones más sencillas de ver, pero la herramienta permite realizar visualizaciones en movimiento, con cambios entre diferentes dibujos y formas, o incluso, enlazando diferentes visualizaciones en una misma pestaña donde pudiéramos elegir que queremos ver, pero que aquí es imposible de enseñar su funcionamiento.

Como se puede intuir, estas visualizaciones pueden ayudar a la mejora del aprendizaje. Se pueden ver, de una manera fácil y sencilla, muchísimos datos de los usuarios, y con ellos, se pueden analizar comportamientos o fallos en el sistema de enseñanza, lo cual, es el objetivo principal de este TFG.

CAPÍTULO 7

CONCLUSIONES Y TRABAJOS

FUTUROS

7. Conclusiones y trabajos futuros

En este capítulo se enumerarán las conclusiones del TFG y se comentaran posibles mejoras futuras sobre el trabajo realizado.

7.1 Conclusiones

El motivo de este TFG era desarrollar una nueva funcionalidad en la plataforma de edX con la que poder ver información útil para los tutores, e incluso para los alumnos.

Después de analizar toda la información disponible de los cursos y los usuarios, se ve claramente el potencial que tendría esta nueva funcionalidad, pudiendo crear estadísticas y gráficas de casi cualquier cosa que valdrían para mejorar la experiencia de los usuarios y el aprendizaje.

También se ven fallos y cosas a mejorar dentro de la aplicación. Existen muchos campos en las bases de datos que se encuentran obsoletos o que ya no se utilizan, pudiendo ser estos bastante útiles para algunas consultas. Al igual que se echan en falta otros campos o la mejora de algunos para hacerlos más accesibles y más fáciles de utilizar.

En general, se ve que es una aplicación que está en constante desarrollo con aportaciones de mucha gente y eso, a veces se nota en problemas de instalación, compatibilidad y en información desfasada, pero con un propósito muy positivo y con mucho potencial de mejora.

En el aspecto de las visualizaciones, la herramienta utilizada da una inmensa libertad para crear gráficas a gusto de todos. Su uso no es difícil y las opciones que da son muy grandes, dando una relación muy positiva de dificultad-resultado.

Finalmente, se puede concluir que los objetivos del TFG se han cumplido, realizando todas las consultas necesarias y realizando todas las visualizaciones que se

querían hacer, haciendo posible el análisis de los datos y su uso para la mejora del aprendizaje.

7.2 Trabajos futuros

Aunque se han cumplido los objetivos del TFG se podrían realizar algunas mejoras en el futuro o desarrollarlo más:

- **Nuevas visualizaciones:** lo primero que se podría hacer sería realizar nuevas consultas y visualizaciones para ampliar más el rango de gráficas posibles.
- **Nuevas tablas:** se podrían crear nuevas tablas en las bases de datos con nueva información que fuera útil para el punto anterior.
- **Integración:** todas estas visualizaciones se podrían incluir, directamente, en una pestaña nueva en todos los cursos.
- **Gamificación:** con todo lo anterior se podrían llegar a crear una serie de hitos en los que, los usuarios, tendrían que realizar ciertas acciones para conseguir todos los hitos. Sería otra manera de analizar el comportamiento de los alumnos.

CHAPTER 8

CONCLUSIONS AND FUTURE WORKS

8. Conclusions and future works

This chapter will list the conclusions of the TFG and discuss possible future improvements in the work done.

8.1 Conclusions

The aim of this TFG was to develop a new functionality in the edX platform with which it is able to see useful information for the tutors, and even for the students.

After analyzing all the available information of the courses and the users, it is clear the potential that this new functionality would have, being able to create statistics and graphs of almost anything that would be worth to improve the users and teaching experience.

It is also seen the failures and things to improve inside the application. There are many fields in databases that are obsolete or are not longer used, which can be quite useful for some queries. Just as it will be missed other fields or the improvement of some of them to make them more accessible and easier to use.

In general, it is seen that is an application that is constantly developing with the contribution of many people and that, sometimes it is notice in set up, compatibility and information's problems, but with a very positive purpose and with much improvement potential.

In the aspect of the visualizations, the tool used gives an immense freedom to create graphs. Its use is not difficult and the options are very large, giving a very positive difficulty-result relationship.

Finally, it can be concluded that the project objectives have been fulfilled, making all the necessary queries and performing all the displays that were wanted to do, making possible the analysis of the data and its use for the improvement of the learning.

8.2 Future works

Although the TFG objectives have been reached, some improvements could be made in the future or make more developments:

- **New displays:** The first thing that could be done would be to make new queries and displays to extend the range of possible graphs.
- **New tables:** new tables could be created in the databases with new information that would be useful for the previous point.
- **Integration:** All these displays could be included directly in a new tab in all the courses.
- **Gamification:** With all of the above, It could be reached in create different objectives in which, the users, would have to perform certain actions to achieve all the objectives. It would be another way of analyzing students' behavior.

CAPÍTULO 9

ANEXOS

9. Anexos

En este capítulo, se precisan todos los aspectos referentes a la gestión del proyecto, la planificación del mismo, los medios técnicos utilizados, el marco regulador y el análisis económico.

9.1 Planificación

A continuación, se desgranará la planificación del TFG en su totalidad, con fechas y duración de cada tarea realizada.

<u>Planificación</u>			
Actividad	Inicio	Fin	Nº días
Trabajo Fin de Grado	31/10/2016	21/06/2017	233
Instalación plataforma	31/10/2016	26/12/2016	57
Documentación edX	31/10/2016	30/11/2016	31
Instalación plataforma	07/11/2016	26/12/2016	50
Creación datos	27/12/2016	08/03/2017	74
Creación usuarios	27/12/2016	02/01/2017	7
Creación del curso	03/01/2017	29/01/2017	27
Creación de datos en foros	30/01/2017	08/03/2017	38
Documentación y diseño BBDD	06/02/2017	06/04/2017	60
Documentación BBDD edX	06/02/2017	12/03/2017	35
Documentación MySQL	12/02/2017	01/03/2017	18

<u>Planificación</u>			
Actividad	Inicio	Fin	Nº días
Consultas MySQL	20/02/2017	17/03/2017	26
Documentación MongoDB	27/02/2017	26/03/2017	28
Consultas MongoDB	06/03/2017	06/04/2017	32
Creación visualizaciones	07/04/2017	05/06/2017	59
Documentación D3.js	07/04/2017	04/05/2017	28
Realización visualizaciones	23/04/2017	05/06/2017	44
Análisis de resultados	22/05/2017	15/06/2017	25
Realización memoria	29/05/2017	21/06/2017	24
Estructuración	29/05/2017	02/06/2017	5
Redacción	01/06/2017	21/06/2017	21

Tabla 2 - Planificación TFG

Para ilustrar estos datos con una escala de tiempos, se ha creado un diagrama de Gantt, donde el tiempo utilizado es el número de semana del año:

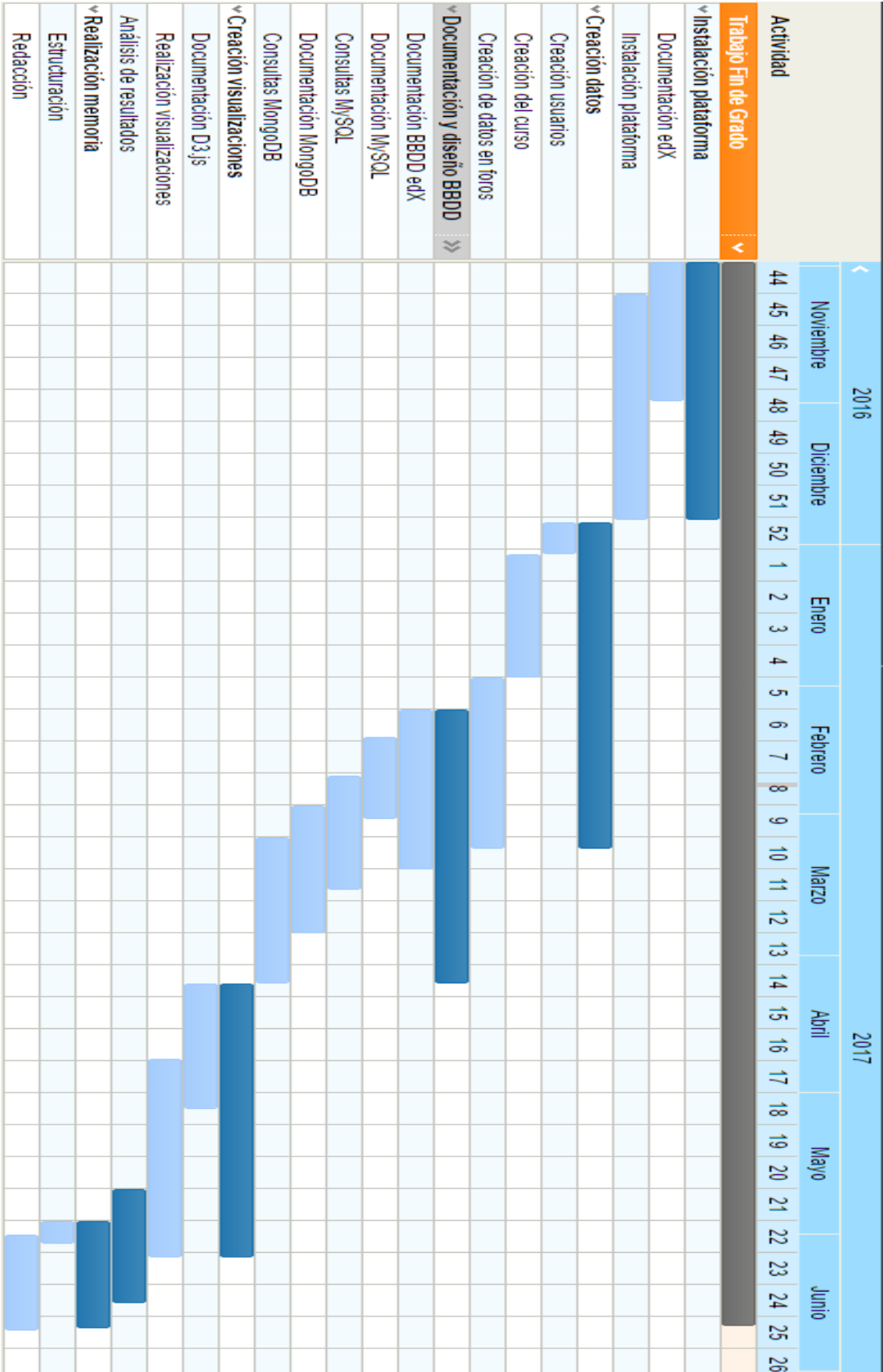


Tabla 3 - Diagrama de Gantt

9.2 Medios técnicos empleados

Para este apartado, especificaremos tanto el hardware como el software utilizado para realizar el TFG al completo.

Hardware

- Ordenador sobre el que se instala la aplicación edX:
 - Ordenador portátil con Ubuntu 14.04 LTS
 - Procesador Intel Core i3-4000M CPU @ 2.40GHz x 4
 - Memoria RAM de 3,8GiB
- Ordenador utilizado para el estudio general y la redacción de la memoria:
 - Ordenador portátil con Windows 8.1
 - Procesador Intel® Core™ i3-3120M CPU @ 2.50GHz x 4
 - Memoria RAM de 6GB

Software

- LibreOffice: paquete de software de oficina, libre y de código abierto desarrollado por The Document Foundation, el cual, ha sido utilizado para la redacción de la memoria.
- Google Chrome: navegador web desarrollado por Google, utilizado para el estudio y para la visualización de los datos.
- Mozilla Firefox: navegador web libre y de código abierto, utilizado para la creación de los datos y del curso de edX.
- Gedit: editor de textos compatible con UTF-8, utilizado para almacenar las consultas de las BBDD.

- Oracle VM VirtualBox: software de virtualización, necesario para la instalación y el uso de la plataforma.
- Vagrant: herramienta para la creación y configuración de entornos de desarrollo virtualizados, necesario para la instalación y el uso de la plataforma.
- Cliente nfsd: protocolo de nivel de aplicación, necesario para el funcionamiento de la plataforma.
- DevStack: script que crea rápidamente un entorno de desarrollo OpenStack, necesario para el funcionamiento de la plataforma.
- MySQL: sistema de gestión de bases de datos relacional en el que se guardan datos de la aplicación.
- MongoDB: sistema de gestión de bases de datos no relacional en el que se guardan datos de la aplicación.
- D3.js: librería de Javascript utilizada para producir infogramas dinámicos e interactivos en navegadores web a partir de datos obtenidos de las bases de datos.
- Servidor HTTP Apache: servidor web HTTP de código abierto, utilizado para poder ver las visualizaciones.
- HTML: lenguaje de marcado para la elaboración de páginas web, utilizado para poder mostrar las visualizaciones.
- Php: lenguaje de programación diseñado para el desarrollo web de contenido dinámico, utilizado para poder mostrar las visualizaciones.

9.3 Marco regulador

Open edX tiene distribuido todo su código en varios repositorios de GitHub, y cada uno está bajo la licencia AGPL o Apache.

AGPL es una licencia “copyleft” (práctica que consiste en el ejercicio del derecho de autor con el objetivo de permitir la libre distribución de copias y versiones modificadas de una obra u otro trabajo, exigiendo que los mismos derechos sean preservados en las versiones modificadas) mantenida por la Free Software Foundation.

Permite el uso y el uso compartido de: código modificado, código no modificado y la modificación del mismo. Por lo tanto, si se modifica la plataforma Open edX, se deben compartir las modificaciones con el resto.

Apache Software License es una licencia de software libre permisiva creada por la Apache Software Foundation (ASF). En este caso, no es una licencia copyleft, ya que no se requiere la redistribución del código fuente cuando se distribuyen versiones modificadas.

Permite al usuario del software la libertad de usar el software para cualquier propósito, bajo los términos de la licencia, sin preocuparse de las “royalties”. Tiene el mismo uso que la AGPL, pero permite compartir las modificaciones bajo otra licencia.

Cada categoría de Open edX se rige bajo una licencia. El código esencial de la plataforma utiliza AGPL, para garantizar que los cambios sean compartidos, y que así, sea realmente una plataforma abierta y donde todos se benefician de las mejoras. Esto incluye la plataforma edX, la configuración de edX y edX-ora2.

Para algunos repositorios donde prima el uso amplio a las mejoras compartidas, como Xblock, se utiliza la licencia Apache. Las herramientas que utiliza edX también utiliza la licencia Apache, ya que no tienen nada que ver con edX y no forman parte de la plataforma.

Para el resto de información que no es código, como los datos de los usuarios de la plataforma, se rigen bajo unos términos de uso, servicio y privacidad, a parte de un código de honor, los cuales permiten el uso de los datos para investigación o la

exposición de ciertos datos para diversos propósitos, todo ello aceptado por todos los usuarios al crear su perfil.

En el caso que nos concierne, la plataforma que utilizamos es local, no tenemos acceso a datos de terceros ni utilizamos datos de terceros. El código lo puede utilizar cualquier persona gracias a las licencias AGPL y Apache, y no es necesario compartirlo, aunque los datos de este TFG podrían incluirse en un proyecto de la UC3M que ya comparte información con el resto de desarrolladores.

9.4 Análisis económico

En este apartado se detallará el análisis económico del Trabajo Fin de Grado:

- **Autor:** David López Móriz
- **Departamento de Ingeniería Telemática**
- **Título:** Gamificación en la plataforma edX
- **Duración:** 7.5 meses
- **Costes indirectos:** 18 %
- **Presupuesto:** ver tabla a continuación

Concepto	Cantidad	Coste (€)	Depreciación	Total (€)
Ordenador portátil con Windows	1	800	60 meses	100€
Ordenador portátil con Ubuntu	1	450	60 meses	56,25€
Graduado en Ing. Telemática	1 (7,5meses)	2,111,81€	-	15.838,57€
Ingeniero Senior	1 (1,5meses)	4.166,67€	-	6.250,01€
Total costes directos	-	-	-	22.244,83€
Costes indirectos (20%)	-	-	-	4.448,97€
Total	-	-	-	26.693,80€

Tabla 4 - Presupuesto del TFG

Bibliografía

[1] Definición e historia de educación a distancia

https://es.wikipedia.org/wiki/Educación_en_línea

[2] Definición e historia de los MOOC.

<http://mooc.es/que-es-un-mooc/>

[3] Más info sobre los MOOC.

https://es.wikipedia.org/wiki/Massive_Open_Online_Course

[4] Plataforma de cursos MOOC sobre el que se desarrolla el TFG.

<https://www.edx.org/>

[5] Documentation for edx.org and the Open edX Community.

<http://docs.edx.org/>

[6] Installing, Configuring, and Running the Open edX Platform

<http://edx.readthedocs.io/projects/edx-installing-configuring-and-running/en/latest/index.html>

[7] Instalación de la plataforma.

“Running Devstack”

<https://openedx.atlassian.net/wiki/display/OpenOPS/Running+Devstack>

[8] Información de los foros de la plataforma.

“Discussion Forums Data”

http://devdata.readthedocs.io/en/latest/internal_data_formats/discussion_data.html

[9] “The MongoDB 3.4 Manual”

<https://docs.mongodb.com/manual/>

[10] Manual de MySQL

<https://dev.mysql.com/doc/refman/5.7/en/preface.html>

[11] Información de los usuarios de los cursos.

“User Info and Learner Progress Data”

http://devdata.readthedocs.io/en/latest/internal_data_formats/sql_schema.html#

[12] Grupos de Google para la ayuda sobre fallos.

<https://groups.google.com/forum/?hl=en#!forum/edx-code>

[13] D3.js – Data-Driven Documents.

<https://d3js.org/>

[14] Galería github de D3.

<https://github.com/d3/d3/wiki/Gallery>

[15] Documentación php.

<http://php.net/>

[16] Ayuda para HTML.

<http://es.html.net/tutorials/html/>

[17] Ayuda para código de HTML.

https://www.w3schools.com/tags/tag_code.asp

[18] Instalación servidor Apache.

<https://httpd.apache.org/>

[19] Información de Learning Analytics.

<http://www.oei.es/historico/divulgacioncientifica/?Learning-analytics-instrumento>

[20] Más información de Learning Analytics.

https://es.wikipedia.org/wiki/Learning_analytics

[21] Herramienta online para realizar diagramas de Gantt.

<https://www.tomsplanner.es/>

